IN THE CLAIMS

Please amend the claims as follows.

1. (Previously Presented) A system for takeover of a Transport Control Protocol (TCP) connection by a second server from a first server comprising:

a structure for the first and second servers adapted to access shared state information with respect to the connection, wherein the structure includes an identification of an application layer protocol being used for communication by an application;

a device for comparing a data packet sequence number of an acknowledgement byte received by the second server with a sequence number related to the shared state information; and

a device for recreating the connection within the second server based upon the compared sequence numbers.

2. (Original) The system as set forth in claim 1 wherein the device for recreating includes an application program interface (API) for communicating with a plurality of protocols in the second server and providing a ready signal in response to a successful comparison by the device for comparing.

3. (Original) The system as set forth in claim 1 further comprising a connection checkpoint application program interface (API) for communicating with each of a plurality of protocols in the first server and for causing each of the plurality of protocols to append relevant state information to a data block passed to each of the plurality of protocols with respect to the connection so as to provide the relevant state information to the shared state information.

4. (Original) The system as set forth in claim 3 wherein the connection checkpoint API is adapted to bundle connection information with respect to a protocol having a plurality of related connections, the related connections involving both TCP protocol and User Datagram Protocol (UDP).

5. (Original) The system as set forth in claim 4 wherein the protocol having a plurality of related connections comprises a protocol having a control connection and a data connection.

6. (Original) The system as set forth in claim 5 wherein one or more of the data connections are carried over UDP or another non-TCP transport protocol.

7. (Original) The system as set forth in claim 6 wherein data packets on one or more of the data connections are adapted to be transmitted to one or more IP-multicast groups.

8. (Original) The system as set forth in claim 1 wherein the shared state information includes an identifier of the first server, the time at which the state information is gathered, a source IP address, a source TCP port, a destination IP address, a destination TCP port, an application layer protocol with respect to the connection, an initial packet sequence number for the source, an initial packet sequence number for the destination, a current packet sequence number for the source, a current packet sequence number for the destination, and application layer information including the TCP sequence number for the first byte of an object and an identifier for the object.

9. (Previously Presented) A method for takeover of a Transport Control Protocol (TCP) connection by a second server from a first server comprising:

generating shared state information with respect to the connection for access by the first server and the second server, wherein the shared state information includes an identification of an application layer protocol being used for communication of an application;

comparing a received data byte sequence number from an acknowledgement byte received by the second server with a sequence number related to the shared state information; and

recreating the connection within the second server based upon the compared sequence numbers.

10. (Original) The method as set forth in claim 9 wherein recreating the connection includes communicating with a plurality of protocols in the second server to provide a ready signal in response to a successful comparison of the sequence numbers.

11. (Original) The method as set forth in claim 9 further comprising performing a connection checkpoint with an application program interface (API) so as to communicate with each of a plurality of protocols in the first server and so as to cause each of the plurality of protocols to append relevant state information to a data block passed to each of the plurality of protocols with respect to the connection, and to thereby provide the relevant state information to the shared state information.

12. (Original) The method as set forth in claim 11 further comprising relating, by the API, the compared sequence numbers in conjunction with the ready signal to a byte sequence number in an object referenced in the shared state information.

13. (Original) The method as set forth in claim 12 in which the relating includes resuming sending the data associated with an object over the restarted TCP connection.

14. (Original) The method as set forth claim 11 further comprising a network protocol stack including the plurality of protocols and checkpoint information based upon the connection checkpoint, and using an application program interface (API) on the second server to notify each of the protocols in the network stack to use the checkpoint information to thereby create an "unready" connection.

15. (Original) The method as set forth in claim 11 wherein the API is adapted to bundle connection information with respect to a protocol having a plurality of related connections.

16. (Original) The method as set forth in claim 15 wherein the protocol having a plurality of related connections comprises a protocol having a control connection and a data connection.

17. (Original) The method as set forth in claim 16 wherein one or more of the data connections are carried over UDP or another non-TCP transport protocol.

18. (Original) The method as set forth in claim 17 wherein data packets on one or more of the data connections are adapted to be transmitted to one or more IP-multicast groups.

19. (Previously Presented) The method as set forth in claim 9 wherein the shared state information includes an identifier of the first server, a time at which the shared state information is received, a source IP address, a source TCP port, a destination IP address, a destination TCP port, the application layer protocol with respect to the connection, an initial packet sequence number for the source, an initial packet sequence number for the destination, a current packet sequence number for the source, a current packet sequence number for the destination, and application layer information.

20. (Original) The method as set forth in claim 9 wherein the recreating includes assignment of the second server to takeover the connection based upon a detecting a failure or overburdening of the first server.

21. (Previously Presented) Functional data embodied in one or more computer-readable media for takeover of a Transport Control Protocol (TCP) connection by a second server from a first server, the functional data comprising:

a device for generating shared state information with respect to the connection for access by the first server and the second server, wherein the shared state information includes an identification of an application layer protocol being used for communication by an application;

a device for comparing a data packet sequence number of an acknowledgement byte received by the second server with a sequence number related to the shared state information; and

a device for recreating the connection within the second server based upon the compared sequence numbers.

22. (Canceled)